

Aula 12 – Gráfico

Disciplina: Programação Web

Prof. Allbert Velleniche de Aquino Almeida

E-mail: allbert.almeida@fatec.sp.gov.br

Site: <http://www.allbert.com.br>



/allbert.almeida

Objetivo

- O objetivo da aula é implementar a construção de uma aplicação ASP.net MVC utilizando a API Charts do Google para visualização de gráficos. Essa aula atende a outros tipos de arquitetura de projeto como webforms por exemplo

Conceito

- A Google Chart API

<https://developers.google.com/chart/> permite a criação de gráficos de forma dinâmica e online. Com ela é possível gerar gráficos nos seguintes tipos: Bar chart, Line chart, Radar chart, Scatter plot, Venn diagram, Pie chart, Google-o-meter, Maps e QR-Code.

Exemplo

- Vamos construir uma aplicação e utilizaremos uma classe “Vendas” inserindo dados manualmente em um “list”, para simular os dados vindo do BD, e aplicaremos agrupamentos para geração dos gráficos.

Exemplo

- Para essa solução aplicaremos três tipos de gráficos: pizza, coluna e barra, porém a uma quantidade variada de tipos de gráficos que podem ser utilizados.

Vamos a Prática

- Abra o Visual Studio >> Novo Projeto. Procure o tipo de solução dentro da linguagem C# >> Web >> Aplicativo Web ASP.NET (.NET Framework). Nomeie a sua aplicação e escolha o caminho onde a aplicação será salva.

Classe Vendas

- Agora, adicionaremos a classe "Vendas" e o método "listar vendas" com um conjunto de vendas para simulação.

Criar uma classe Vendas

```
public class Vendas {
    public string Produto { get; set; }
    public string Categoria { get; set; }
    public DateTime Data { get; set; }
    public Double Valor { get; set; }

    public static List<Vendas> ListaVendas() {
        List<Vendas> lst = new List<Vendas>();
        lst.Add(new Vendas { Categoria = "Calçados", Data =
DateTime.Parse("2020-05-21"), Produto = "Produto1", Valor = 100.00 });
        lst.Add(new Vendas { Categoria = "Vestuário", Data =
DateTime.Parse("2020-05-21"), Produto = "Produto2", Valor = 125.36 });
        lst.Add(new Vendas { Categoria = "Vestuário", Data =
DateTime.Parse("2020-04-21"), Produto = "Produto2", Valor = 130.45 });
        lst.Add(new Vendas { Categoria = "Calçados", Data =
DateTime.Parse("2020-04-21"), Produto = "Produto1", Valor = 110.77 });
        lst.Add(new Vendas { Categoria = "Calçados", Data =
DateTime.Parse("2020-04-21"), Produto = "Produto3", Valor = 100.21 });
        lst.Add(new Vendas { Categoria = "Vestuário", Data =
DateTime.Parse("2020-03-21"), Produto = "Produto2", Valor = 140.33 });
        lst.Add(new Vendas { Categoria = "Calçados", Data =
DateTime.Parse("2020-03-21"), Produto = "Produto1", Valor = 90.55 }); ;
        return lst;
    }
}
```


Classe Funcoes

- Vamos adicionar uma classe de funções, nessa pode conter todos os métodos auxiliares que sirva de forma geral para o projeto. Adicione na pasta “Models” a classe “Funcoes.cs”, insira o código de geração dos gráficos e as referências:

Na Classe Funcoes – Gráfico Pizza

```
public static string GerarGraficoPizza(string titulo, string dados) {
    //dados: ['Work', 11],['Eat', 2],['Commute', 2],['Watch TV', 2],['Sleep', 7]
    string graf = @"<script type='text/javascript'
src='https://www.gstatic.com/charts/loader.js'></script>
    <script type='text/javascript'>
        google.charts.load('current', {packages:['corechart']});
        google.charts.setOnLoadCallback(drawChart);
        function drawChart() {
            var data = google.visualization.arrayToDataTable([
                [' ', ''],
                " + dados + @"
            ]);

            var options = {
                title: '" + titulo + @"',
                is3D: true, };

            var chart = new
google.visualization.PieChart(document.getElementById('piechart_" +
titulo.Replace(" ", "") + @""));
            chart.draw(data, options);
        }
    </script>
    <div id='piechart_" + titulo.Replace(" ", "") + @"' style='min-
height: 500px;'></div>";
    return graf;
}
```

Usar o gráfico

- Com os dados de vendas já inseridos no método e as função de geração de gráfico já realizadas, no arquivo "HomeController.cs" na pasta "Controller", vamos adicionar na action "Index" para pegar os dados;

Action Index (Home Controller)

```
public ActionResult Index() {
    var resultado1 = Vendas.ListaVendas().GroupBy(x =>
x.Produto).Select(g => new { g.Key, TotalVendas = g.Sum(x =>
x.Valor) });
    string dados = "";
    // dados: ['Work', 11],['Eat', 2],['Commute', 2],['Watch
TV', 2],['Sleep', 7]
    foreach (var item in resultado1) {
        dados +=
"['"+item.Key+"',"+item.TotalVendas.ToString().Replace(",","."")+"],";
    }
    dados = dados.Substring(0, dados.Length - 1);
    ViewBag.GraficoPizza = Funcoes.GerarGraficoPizza("Total
de Vendas por Produto", dados);
    return View();
}
```

View Index - Home

- A "viewbag" gerada na action, deve ser inserida na "Views >> Home >> Index.cshtml" dentro da linha após o "jumbotron". Para exibir os dados é importante a renderização em html:

```
<div class="col-md-12">  
    @Html.Raw(ViewBag.GraficoPizza)<br /><br />  
</div>
```

Compilar a solução

- Compilar a solução e realizar os testes de visualização do gráfico;

Na Classe Funcoes Gráfico Barra/Coluna

```
public static string GerarGraficoBarraColuna(string titulo, string subtitulo, string dados, bool barra) {
    /*    Dados
        ['Year', 'Sales', 'Expenses', 'Profit'],
        ['2014', 1000, 400, 200],
        ['2015', 1170, 460, 250],
        ['2016', 660, 1120, 300],
        ['2017', 1030, 540, 350]
    */
    string tipo = "";
    if (barra)
        tipo = "bars: 'horizontal'";
    string graf = @"<script type='text/javascript' src='https://www.gstatic.com/charts/loader.js'></script>
        <script type='text/javascript'>
            google.charts.load('current', {'packages':['bar']});
            google.charts.setOnLoadCallback(drawChart);

            function drawChart() {
                var data = google.visualization.arrayToDataTable([
                    " + dados + @""]);
                var options = {
                    chart: {
                        title: '" + titulo + @"',
                        subtitle: '" + subtitulo + @'", },
                    " + tipo + @"
                };
                var chart = new google.charts.Bar(document.getElementById('barchart_' +
titulo.Replace(" ", "") + @''));
                chart.draw(data, google.charts.Bar.convertOptions(options));
            }
        </script>
        <div id='barchart_' + titulo.Replace(" ", "") + @"' style='min-height: 500px;'></div>";
    return graf;
}
```

Usar o gráfico

- Agora iremos adicionar na mesma action "Index()" as outras gerações de gráficos, antes de "return View()":

Action Index (Home Controller)

```
var resultado2 = Vendas.ListaVendas().GroupBy(x => x.Categoria).Select(g => new { g.Key, TotalVendas = g.Sum(x => x.Valor) });
string dadostopo2 = "[";
string dadoscorpo2 = "[" + "Categorias";
foreach (var item in resultado2)
{
    dadostopo2 += "," + item.Key + "";
    dadoscorpo2 += "," + item.TotalVendas.ToString().Replace(",", ".");
}
dadostopo2 += "],";
dadoscorpo2 += "];";
ViewBag.GraficoColuna = Funcoes.GerarGraficoBarraColuna("Total de Vendas por Categoria","", dadostopo2+dadoscorpo2,false);

//-----

var resultado3 = Vendas.ListaVendas().GroupBy(x => new { x.Data.Month, x.Categoria }).Select(g => new { g.Key.Month,g.Key.Categoria, TotalVendas = g.Sum(x => x.Valor) }).OrderBy(x=>x.Month).ThenBy(x=>x.Categoria);

var categorias = resultado3.OrderBy(x => x.Categoria).Select(x => x.Categoria).Distinct().ToList();
var meses = resultado3.OrderBy(x => x.Month).Select(x => x.Month).Distinct().ToList();
string dadostopo3 = "[" + "Mês'," + string.Join(", ", categorias.ToArray()) + "],";
string dadoscorpo3 = "";
foreach (var m in meses)
{
    dadoscorpo3 += "[" + DateTimeFormatInfo.CurrentInfo.GetMonthName(m)+"',";
    foreach (var item in resultado3.Where(x=>x.Month==m).ToList())
    {
        dadoscorpo3 += item.TotalVendas.ToString().Replace(",", ".") + ",";
    }
    dadoscorpo3 = dadoscorpo3.Substring(0, dadoscorpo3.Length - 1);
    dadoscorpo3 += "],";
}
dadoscorpo3 = dadoscorpo3.Substring(0, dadoscorpo3.Length - 1);
ViewBag.GraficoBarra = Funcoes.GerarGraficoBarraColuna("Total de Vendas","Vendas por período", dadostopo3 + dadoscorpo3, true);
```

View Index - Home

- Após o gráfico de pizza as outras "ViewBags" para visualização dos gráficos de coluna e barra:

```
<div class="col-md-6">  
    @Html.Raw(ViewBag.GraficoColuna)  
</div>  
<div class="col-md-6">  
    @Html.Raw(ViewBag.GraficoBarra)  
</div>  
<br /><br />
```

Compilar a solução

- Compilar a solução e realizar os testes de visualização do gráfico;